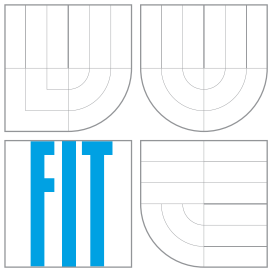
BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

ROBUST SPEAKER VERIFICATION

ROBUSTNÍ ROZPOZNÁVÁNÍ MLUVČÍHO

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR
AUTOR PRÁCE

JÁN PROFANT

SUPERVISOR
VEDOUCÍ PRÁCE

Ing. PAVEL MATĚJKA, Ph.D.

BRNO 2016

Abstract

The goal of this paper is to analyze the impact of codec degraded speech on a state-of-the-art speaker recognition system. Two feature extraction techniques are analyzed — Mel Frequency Cepstral Coefficients (MFCC) and the state-of-the-art system using Bottleneck features together with MFCC. Speaker recognition system is based on i-vector and Probabilistic Linear Discriminant Analysis (PLDA). We compared scenarios where PLDA is trained only on clean data, then system where we added also noise and reverberant data, and at last, codec degraded speech. We evaluated the systems on the matched conditions (data from the same codec are seen with PLDA) and also mismatched conditions (PLDA does not see any data from the tested codec). We experimented also with recently introduced technique for channel adaptation — Within-class Covariance Correction (WCC). We can see clear benefit of adding transcoded data to PLDA or WCC (with approximately same gain) for both tested conditions (matched and mismatched).

Abstrakt

Cílem této práce je analyzovat úspěšnost systému rozpoznávaní mluvčího na nahrávkach degradovaných různym telefonním přenosovým kanálem. Použili jsme dva způsoby extrakce příznaků - Mel Frequency Cepstral Coefficients (MFCC) a moderní systém, který spojuje Bottleneck příznaky spolu s MFCC. Systém rozpoznávání mluvčího je založen na i-vektorech a Pravděpodobnostní Lineární Diskriminační Analýze (PLDA). Porovnali jsme scenáře, kde je PLDA trénovaná jen na čisté řeči, poté systém kde jsme přidali data s hlukem a reverberací a nakonec, data degradované kodekem. Vyhodnotili jsem systémy za rovnakých podmínek (data ze stejného kodeku byli také v trénování PLDA) a také za rozdílnych podmínek (data ze stejného kodeku resp. rodiny kodeků nebyli v trénování PLDA). Také jsme experimentovali s nedávno představenou technikou na adaptaci kanálu — Within-class Covariance Correction (WCC). Můžeme jednoznačně vidět zlepšení úspěšnosti přidáním degradovaných dat do PLDA resp. WCC (s přibližně stejným výsledkem) pro obě naše testované podmínky.

Keywords

speaker verification, Probabilistic Linear Discriminant Analysis, Within-class Covariance Correction, i–vector

Klíčová slova

rozpoznávání mluvčího, Pravděpodobnostní Lineární Diskriminační Analýza, Within-class Covariance Correction, i–vektor

Reference

PROFANT, Ján. *Robust Speaker Verification.* Brno, 2016. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Matějka Pavel.

Robust Speaker Verification

Declaration

Hereby I declare that this bachelor's thesis was prepared as an original author's work under supervision of Mr. Pavel Matějka. The supplementary information was provided by Mr. Ondřej Glembek and Mr. Oldřich Plchot. All the relevant information sources, which were used during preparation of this thesis are properly cited and included in the list of references.

.....
Ján Profant
May 14, 2016

Acknowledgements

I would like to thank my supervisor Pavel Matějka and also thank Ondřej Glembek, Oldřich Plchot and Filip Povolný for their help.

Contents

Chapter 1

Introduction

I-vector based systems have recently become the state-of-the-art framework in Speaker Recognition. I-vector approach was introduced in speaker recognition [5], but has been widely used in multiple fields of speech processing, such as age estimation [2, 8], emotion detection [16], speech recognition [13, 24] and also language recognition [18]. They provide an elegant way of reducing the large-dimensional variable-length input data to a small fixed-dimensional feature vector while retaining most of the relevant information. The objective in speaker verification calls for robust extraction of the relevant speaker information. However, an i-vector contains not only the speaker information, which corresponds to wanted variability in the i-vector space, but also all kinds of unwanted information [5] that is commonly referred to as *channel variability*. There are various techniques to deal with these two types of variability which all aim at suppressing as much of the channel variability and emphasizing as much of the speaker variability as possible.

Compressed audio plays a significant role in mobile communications, Voice Over Internet Protocol (VOIP), archival audio storage, gaming communications and also internet streaming audio. In most of these tasks, there is a widespread use of lossy speech coders. The purpose of speech coders is to compress the speech signal to reduce the number of bits needed for transmission, while maintaining the intelligibility of speech once decoded.

In Speaker Recognition, the distortion introduced by speech coders may have a significant negative impact on the performance — therefore, the analysis of codec-related degradation and the development of robust techniques against this degradation is very actual topic. Ideal speaker verification system should achieve same results on any codec variation.

There have been a number of papers investigating the effect of codecs on speaker recognition performance. The paper [20] analyzes the effect of codec distortion on the probabilistic linear discriminant analysis (PLDA) compensation module on systems with three different feature extractions (MFCC, PNCC and MDMC). The paper also explores i-vector fusion as a way to increase robustness to codec distortions.

In my work, I also analyzed the effect of codec distortion on the PLDA compensation module with the state-of-art system [19] reaching twice better results than the Baseline from [20]. Later, I used a Within-Class Covariance Correction (WCC) [11] technique for Linear Discriminant Analysis (LDA) to analyze impact of codec-degraded speech on speaker verification system, I used WCC to improve performance of our system on codec-degraded speech. Finally, I used both techniques together to improve performance of my system on codec-degraded speech.

At last, I analyzed effect of length of audio record on speaker verification system per-

formance, when using exact length of speech extracted from record with Voice Activity Detector. I retrained whole system on different lengths, compared it to Baseline system and also analyzed approach, when length of speech used for training is randomly generated.

Chapter 2

Theoretical Background

2.1 Speaker Recognition

Speaker recognition is the identification of a person from characteristics of voices. No two individuals sound identical because their vocal tract shapes, larynx sizes, and other parts of their voice production organs are different. In addition to these physical differences, each speaker has characteristic manner of speaking, including the use of a particular accent, rhythm, intonation style, pronounciation pattern, or choice of vocabulary.

There are two major applications of speaker recognition technologies and methodologies. If the speaker claims to be of a certain identity and the voice is used to verify this claim, this is called *verification* or *authentication*. On the other hand, *identification* is the task of determining an unknown speaker's identity [26].

Application dictates different speech modalities:

- **Text-dependent** - recognition system knows text spoken by person, knowledge of spoken text can improve system performance.
- **Text-independent** - recognition system does not know text spoken by person, more flexible system but more difficult problem.

2.2 Voice Activity Detection

Voice Activity Detection (VAD) is used in telecommunications, for example, in telephony to detect touch tones and the presence or absence of speech. Detection of speaker activity can be useful in responding to barge-in, for pointing to the end of an utterance in automated speech recognition, and for recognizing a word intended to trigger start of a service, application, event, or anything else that may be deemed useful.

VAD is typically based on the amount of energy in the signal (a signal having more than a threshold level of energy is assumed to contain speech, for example) and in some cases also on the rate of zero crossings, which gives a crude estimate of its spectral content. If the signal has high-frequency components then zero-crossing rate will be high and vice versa. In general, in one aspect, the invention features a method that includes using a subset of values to discriminate voice activity in a signal, the subset of values belonging to a larger set of values representing a segment of speech and the larger set of values being suitable for speech recognition [4].

2.3 Feature Extraction

Speech signal includes many features of which not all are important for speaker discrimination. An ideal feature would:

- have large between-speaker variability and small within-speaker variability
- be robust against noise and distortion
- occur frequently and naturally in speech
- be easy to measure from speech signal
- be difficult to impersonate/mimic
- not be affected by the speaker's health or long-term variations in voice

The number of features should be also relatively low. Traditional statistical models such as the Gaussian Mixture Model (GMM) cannot handle high-dimensional data. The number of required training samples for reliable density estimation grows exponentially with the number of features and the computational savings are also obvious with low-dimensional features [26].

2.3.1 Mel Frequency Cepstral Coefficient

Mel Frequency Cepstral Coefficients (MFCC) are a feature widely used in automatic speech and speaker recognition. Figure 2.1 shows procedure, how to calculate MFCCs [12].

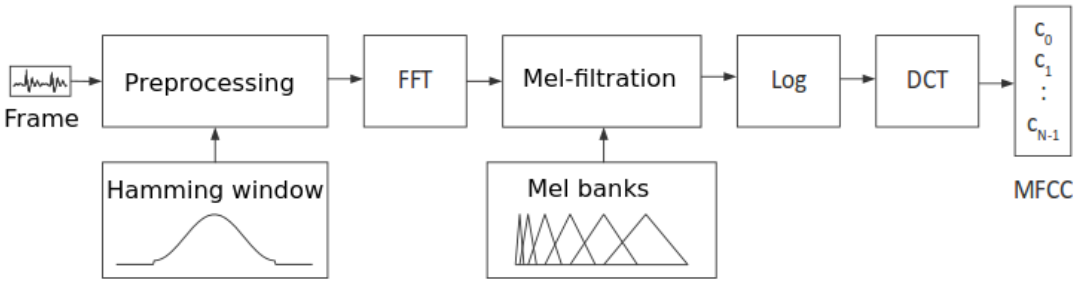


Figure 2.1: *Scheme of calculating MFCCs.*

Here, we can see more detailed description of how to calculate MFCCs according to Figure 2.1:

1. Frame the signal into short frames.
2. For each frame calculate the periodogram estimate of the power spectrum.
3. Apply the mel filterbank to the power spectra, sum the energy in each filter.
4. Take the logarithm of all filterbank energies.
5. Take the DCT of the log filterbank energies.

2.3.2 Bottleneck Features

Bottleneck Neural-Network (BN-NN) refers to such topology of a NN, where one of the hidden layers has significantly lower dimensionality than the surrounding layers. A bottleneck feature vector is generally understood as a by-product of forwarding a primary input feature vector through the BN-NN and reading off the vector of values at the bottleneck layer. We have used a cascade of two such NNs for our experiments. The output of the first network is *stacked* in time, defining context-dependent input features for the second NN, hence the term Stacked Bottleneck Features. The dimensionality of the bottleneck layer was fixed to 80. For more details see [9, 19].

2.4 Gaussian Mixture Model

A Gaussian Mixture Model (GMM) is a generative model that assumes all the data points are generated from a mixture of a finite number of Gaussian (normal) distributions with unknown parameters. Gaussian distributions are important in statistics and are often used in the natural and social sciences to represent real-valued random variables whose distributions are not known.

The probability density of the multivariate Gaussian distribution is:

$$\mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^P|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{2.1}$$

where $\boldsymbol{\mu}$ is the mean and parameter $\boldsymbol{\Sigma}$ is variance matrix with its matrix determinant $|\boldsymbol{\Sigma}|$. Frequently used methods to estimate parameters are maximum a posteriori probability (MAP) and maximum likelihood (ML). MAP estimator chooses class with highest posteriori probability from N classes:

$$arg\ max_\omega\ P(\omega|x) = arg\ max_\omega\ P(x|\omega)\ P(\omega). \tag{2.2}$$

Maximum likelihood is a method of estimating the parameters of a statistical model given data, as follows [17]:

$$\boldsymbol{\Theta}_{ML}^{class} = arg\ max_{\boldsymbol{\Theta}} \prod_{x_i \in class} p(x_i|\boldsymbol{\Theta}). \tag{2.3}$$

where $\boldsymbol{\mu}$ is estimated as

$$\boldsymbol{\mu} = \frac{1}{T}\sum\nolimits_i \boldsymbol{x}_i, \tag{2.4}$$

and covariance matrix $\boldsymbol{\Sigma}$ as

$$\boldsymbol{\Sigma} = \frac{1}{T}\sum\nolimits_i (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T. \tag{2.5}$$

GMM is then a probabilistic generative model,

$$p(\boldsymbol{x}|\Theta) = \sum\nolimits_c \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu_c},\boldsymbol{\Sigma_c})P_c \tag{2.6}$$

where, $\boldsymbol{\Theta} = \{\boldsymbol{P}_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}$ is set of parameters and $\sum_c \boldsymbol{P}_c = \mathbf{1}$.

A GMM is used in speaker recognition applications as a generic probabilistic model for multivariate densities capable of representing arbitrary densities, which makes it well suited for unconstrained text-independent applications [7].

2.5 UBM-GMM Adaptation

A UBM-GMM (Universal Background Model - Gaussian Mixture Model) is a model in a speaker verification system to represent general, person-independent, channel independent feature characteristics to be compared against a model of speaker-specific feature characteristics when making an accept or reject decision. Here, the UBM is a speaker-independent GMM trained with speech samples from a large set of speakers to represent general speech characteristics. The UBM is also used when training the speaker-specific model by acting as a prior model in MAP parameter estimation.

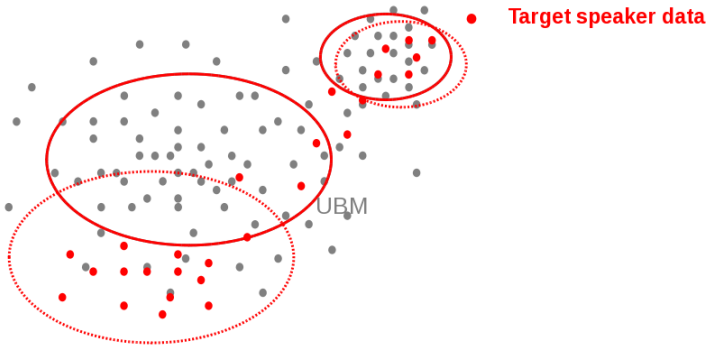


Figure 2.2: *Speaker model adapted from UBM-GMM - target speaker data are adapted to UBM-GMM, mixture weights and covariance matrices are copied from UBM.*

Based on Figure 2.2, it is obvious that only means are adapted using relevance MAP adaptation

$$\mu_c^{MAP} = (\gamma_c \mu_c^{ML} + \tau \mu_c^{UBM})/(\gamma_c + \tau) \tag{2.7}$$

where μ^{UBM} is UBM mean vector μ^{ML} ML retrained on enrollment data, γ_c is occupation counts for c^{th} components and τ is relevance factor (typically between 10 and 20) [23].

2.6 i-vector

The i-vector approach has become state of the art in the speaker verification field [6]. The approach provides an elegant way of reducing large-dimensional input data to a small-dimensional feature vector while retaining most of the relevant information. The technique was originally inspired by the JFA framework [15]. The basic principle is that on some data, we train the i-vector extractor and then for each speech segment, we extract the i-vector as a low-dimensional fixed length representation of the segment. The main idea is that the speaker- and session-dependent supervectors of concatenated GMM means can be modeled as

$$\mathbf{s} = \mathbf{m} + \mathbf{T}\mathbf{x}, \tag{2.8}$$

where $\mathbf{m}$ is the UBM GMM mean supervector, $\mathbf{T}$ is a matrix of bases spanning the subspace covering the important variability (both speaker- and session-specific) in the supervector space, and $\mathbf{x}$ is a standard-normally distributed latent variable. For each observation sequence representing a segment, our i-vector $\boldsymbol{\phi}$ is the MAP point estimate of the latent variable $\mathbf{x}$ [3].

2.7 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order avoid overfitting and also reduce computational costs. In Figure 2.3 we can see, to which direction we want to transfer our data.

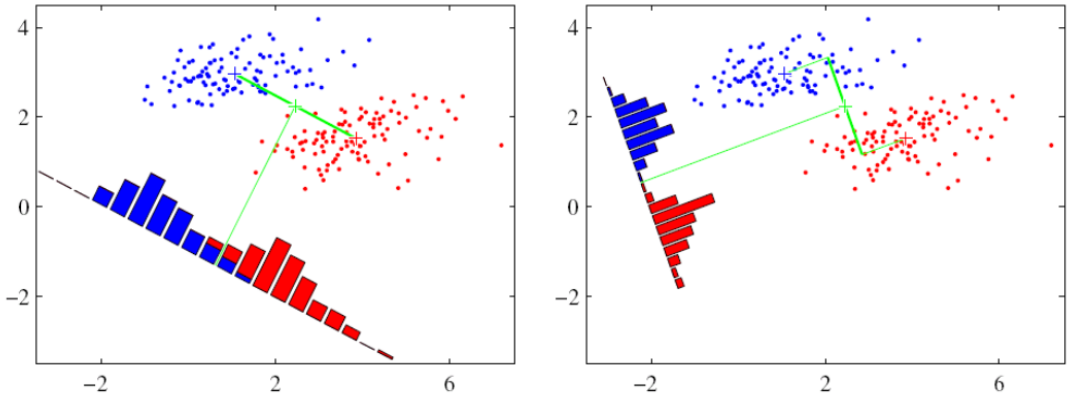


Figure 2.3: *LDA: Maximize distance between* $\boldsymbol{\mu_c}$ *of classes and minimize average variance of classes.*

Let us recall that LDA is based on computing the between-class and within-class covariance matrices Σ_B and Σ_W respectively, whose Maximum-Likelihood (ML) is given as

$$\boldsymbol{\Sigma_B} = \frac{1}{N}\sum_{\boldsymbol{c=1}}^{\boldsymbol{C}} \boldsymbol{N_c}(\boldsymbol{\mu_c} - \boldsymbol{\mu})(\boldsymbol{\mu_c} - \boldsymbol{\mu})^T \tag{2.9}$$

$$\boldsymbol{\Sigma_W} = \frac{1}{N}\sum_{\boldsymbol{c=1}}^{\boldsymbol{C}}\sum_{\boldsymbol{n=1}}^{\boldsymbol{N_c}} (\boldsymbol{\phi_{n,c}} - \boldsymbol{\mu})(\boldsymbol{\phi_{n,c}} - \boldsymbol{\mu})^T \tag{2.10}$$

where $\boldsymbol{\phi_{n,c}}$, the n-th data point in class c, C is number of classes, N_c is number of data-points in class c, μ_c is the mean of the data belonging to class c:

$$\boldsymbol{\mu_c} = \frac{1}{N_c}\sum_{\boldsymbol{n=1}}^{\boldsymbol{N_c}} \boldsymbol{\phi_{n,c}} \tag{2.11}$$

where $\boldsymbol{\phi_{n,c}}$, the n-th data point in class c, and μ is the global mean of the data, computed as

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \boldsymbol{\phi_n} \tag{2.12}$$

LDA emphasizes discrimination of data belonging to different classes and it does so by solving the generalized eigen-value problem:

$$\boldsymbol{\Sigma_B v_m} = \lambda_m \boldsymbol{\Sigma_B v_m} \tag{2.13}$$

with $V = [\boldsymbol{v_1}, \ldots, \boldsymbol{v_{\hat{M}}}]$ for $\hat{M}$ largest eigen-values λ_m, and applying V as

$$\boldsymbol{\phi_{LDA}} = \boldsymbol{V^T}\phi \tag{2.14}$$

Class separability for each basis is often expressed by the Fisher ratio and is equal to the basis corresponding eigen-value [11].

2.8 Probabilistic Linear Discriminant Analysis

To facilitate comparison of i-vectors in a verification trial, the distribution of i-vectors is modeled using a Probabilistic Linear Discriminant Analysis model [22, 14]. First, consider only a special form of PLDA, a *two-covariance model*, in which speaker and inter-session variability are modeled using across-class and within-class full covariance matrices Σ_{ac} and Σ_{wc}. The two-covariance model is a generative linear-Gaussian model, where latent vectors $\mathbf{y}$ representing speakers (or more generally classes) are assumed to be distributed according to prior distribution

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \Sigma_{ac}). \tag{2.15}$$

For a given speaker represented by a vector $\hat{\mathbf{y}}$, the distribution of i-vectors is assumed to be

$$p(\boldsymbol{\phi}|\hat{\mathbf{y}}) = \mathcal{N}(\boldsymbol{\phi}; \hat{\mathbf{y}}, \Sigma_{wc}). \tag{2.16}$$

The ML estimates of the model parameters, $\boldsymbol{\mu}$, Σ_{ac}, and Σ_{wc}, can be obtained using an EM algorithm as in [14]. The training i-vectors come from a database comprising recordings of many speakers (to capture across-class variability), each recorded in several sessions (to capture within-class variability).

2.9 Within-Class Covariance Correction

The anatomy of our recognition system is based on a comparison of a pair of pre-processed i-vectors. The comparison is facilitated via PLDA model. Given a pair of i-vectors, PLDA allows to compute the log-likelihood for the same-speaker hypothesis and for the different-speaker hypothesis. The pre-processing consists of applying LDA to reduce the dimensionality was already discussed in this work. Such processed i-vectors are then followed by global mean and variance normalization, followed by length-normalization. This Section was based on paper [11] with author's permission.

Within-Class Covariance Correction (WCC) is a technique for Linear Discriminant Analysis (LDA) in Speaker Recognition to perform an unsupervised adaptation of LDA to an unseen data domain, and/or to compensate for speaker population difference among different portions of LDA training dataset. It is based on adding extra variability into within-class covariance matrix.

Let us decompose the within-speaker variability as

$$\boldsymbol{\Sigma_{WS}} = \boldsymbol{\Sigma_{BD}} + \boldsymbol{\Sigma_{IS}} \tag{2.17}$$

where $\boldsymbol{\Sigma_{WS}} = \boldsymbol{\Sigma_{BD}}$ is the between-dataset covariance, and $\boldsymbol{\Sigma_{IS}}$ is the intersession covariance, describing an average speaker variability within a dataset and assumed to be shared across datasets.

It can be expressed as a within-class covariance where the classes are pairs of speaker and datasets (d, s):

$$\boldsymbol{\Sigma_{IS}} = \frac{1}{N}\sum_{d=1}^{D}\sum_{s=1}^{S_d}\sum_{n=1}^{N_{d,s}} N_{d,s}(\phi_{n,d,s} - \mu_{d,s})(\phi_{n,d,s} - \mu_{d,s})', \tag{2.18}$$

where D is number of datasets, s is a speaker instance for dataset d.
Other variables have obvious meanings. We can decompose the total variability Σ_T as

$$\boldsymbol{\Sigma_T} = \boldsymbol{\Sigma_{BS}} + \boldsymbol{\Sigma_{WS}} = \boldsymbol{\Sigma_{BS}} + \boldsymbol{\Sigma_{BD}} + \boldsymbol{\Sigma_{IS}} \tag{2.19}$$

Figure 2.4 depicts this situation.

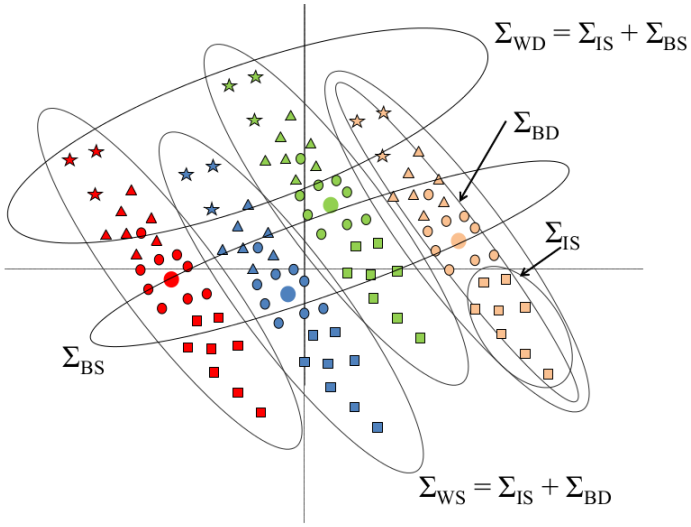


Figure 2.4: *Illustration of decomposition of within-speaker covariance Σ_{WS} into inter-session covariance Σ_{IS}, and between-dataset covariance Σ_{BD}. The within-dataset covariance is decomposed into inter-session covariance and between-speaker covariance. Note that different colors represent different speakers and different shapes represent different datasets.*

It is important to note that, if the speakers do not overlap across datasets, the *speaker* class will effectively (without any change in meaning) be understood as *speaker and dataset* class. As a result, Σ_{WS} will have the same form as Σ_{IS}. The Σ_{BD} term will diminish and will be absorbed by Σ_{BS} , incorrectly emphasizing discriminability power in LDA computation. Note that, multi-modality over datasets does not necessarily have to be a problem if the speakers in the datasets overlap. In this case, the between-dataset variability is correctly included in the within-speaker (channel) covariance.

Assumption is that the directions of dataset shift are exclusive, meaning not carrying any useful speaker information and we can give them very high (negative) importance by scaling the Σ_{BD} up, so that the Fisher ratio in LDA for these directions will be very small. As a consequence, these directions will not be included the final LDA projection. This leads to an update formula

$$\boldsymbol{\Sigma_{WS}^{(new)}} = \boldsymbol{\Sigma_{WS}} + \alpha\boldsymbol{\Sigma_{BD}^{(WCC)}} \tag{2.20}$$

where, Σ_{BD}^{WCC} is the correction matrix estimated as a between-class covariance matrix with datasets as classes. In this case, however, instead of using the ML estimate the number of data in each dataset is equalized, therefore

$$\boldsymbol{\Sigma_{BD}^{(WCC)}} = \frac{1}{C}\sum_{\boldsymbol{c=1}}^{\boldsymbol{C}}(\boldsymbol{\mu_c} - \hat{\boldsymbol{\mu}})(\boldsymbol{\mu_c} - \hat{\boldsymbol{\mu}})^T \tag{2.21}$$

where $\hat{\mu}$ is the equalized (not weighted) average of dataset means:

$$\hat{\mu} = \frac{1}{C}\sum_{c=1}^{C}\mu_c \tag{2.22}$$

2.10 Audio Codecs

An audio codec is a device or computer program capable of coding or decoding a digital data stream of audio. Codec in software is a computer program implementing an algorithm that compresses and decompresses digital audio data according to a given audio file or streaming media audio coding format. The objective of the algorithm is to represent the high-fidelity audio signal with minimum number of bits while retaining the quality. This can effectively reduce the storage space and the bandwidth required for transmission of the stored audio file. For more details about specific codecs see 3GPP [1] or ITU-T [2] standards.

A number of codecs was selected to representative of those currently in widespread use.

2.10.1 AMR

The Adaptive Multi-Rate (AMR, AMR-NB or GSM-AMR) codec is an audio compression format optimized for speech coding. AMR speech codec consists of a multi-rate narrowband speech codec that encodes narrowband (200-3400 Hz) signals at variable bit rates ranging from 4.75 to 12.2 kbit/s with toll quality speech starting at 7.4 kbit/s. AMR was adopted as the standard speech codec by 3GPP in October 1999 and is now widely used in GSM and UMTS and many modern mobile telephone handsets can store short audio recordings in the AMR format. The common filename extension is **.amr* or less used **.3ga*.

The frames contain 160 samples and are 20 milliseconds long. AMR uses various techniques, such as ACELP (Algebraic code-excited linear prediction), DTX (Discontinuous transmission, see 2.10.1), VAD (see 2.2) and CNG (Comfort noise). There are a total of 14 modes of the AMR codec, 8 are available in a full rate channel and 6 on a half rate channel.

These variations of AMR codec were used for audio compression:

- DTX `on`, bitrate 4.75 kbit/s,
- DTX `on`, bitrate 7.40 kbit/s,
- DTX `on`, bitrate 12.20 kbit/s,
- DTX `off`, bitrate 4.75 kbit/s,

[1] http://www.3gpp.org/
[2] http://www.itu.int/en/ITU-T/Pages/default.aspx

DTX

Discontinuous transmission (DTX) is a method of momentarily powering-down, or muting, a mobile or portable wireless telephone set when there is no voice input to the set. This optimizes the overall efficiency of a wireless voice communications system.

In a typical two-way conversation, each individual speaks slightly less than half of the time. If the transmitter signal is switched on only during periods of voice input, the duty cycle of the telephone set can be cut to less than 50 percent. This conserves battery power, eases the workload of the components in the transmitter amplifiers, and frees the channel so that time-division multiplexing can take advantage of the available bandwidth by sharing the channel with other signals.

2.10.2 AMR-WB

Adaptive Multi-Rate Wideband codec is based on Adaptive Multi-Rate encoding, using similar methodology as ACELP [25]. AMR-WB provides improved speech quality due to a wider speech bandwidth of 50-7000 Hz. A common file extension for AMR-WB file format is *.awb*. AMR-WB operates, like AMR (see 2.10.1, with nine different bit rates. The lowest bit rate providing excellent speech quality in a clean environment is 12.65 kbit/s. Higher bit rates are useful in background noise conditions and for music. Also, lower bit rates of 6.60 and 8.85 kbit/s provide reasonable quality, especially when compared to narrow-band codecs. The most often 12.65 kbit/s bitrate is used.

The frequencies from 6.4 kHz to 7 kHz are only transmitted in the highest bitrate mode (23.85 kbit/s), while in the rest of the modes the decoder generates sounds by using the lower frequency data (75-6400 Hz) along with random noise (in order to simulate the high frequency band). All modes are sampled at 16 kHz (using 14-bit resolution) and processed at 12.8 kHz.

These variations of AMR-WB codec were used for audio compression:

- DTX `on`, bitrate 6.60 kbit/s,
- DTX `on`, bitrate 12.65 kbit/s,
- DTX `on`, bitrate 23.05 kbit/s,

2.10.3 G.726

G.726 is an ITU-T ADPCM speech codec standard covering the transmission of voice at rates of 16, 24, 32, and 40 kbit/s. It was introduced to supersede both G.721, which covered ADPCM at 32 kbit/s, and G.723, which described ADPCM for 24 and 40 kbit/s. Sampling frequency of G.726 is 8 kHz. Cisco uses this codec in PSTN switchboards at 32 kbit/s.

Variation with 16 kbit/s bitrate was used.

2.10.4 G.728

G.728 is an ITU-T standard for speech coding operating at 16 kbit/s. It uses LD-CELP (Low-Delay Code Excited Linear Prediction) [25] compression technology. Delay of the codec is only 5 samples (0.625 ms) and it is also reason why G.728 is used in satellite, cellular, and video conferencing systems. It delivers approximately the same quality voice as G.726 with 32 kbit/s bitrate while using only half the bandwidth.

2.10.5 G.729

G.729 is an audio data compression algorithm for voice that compresses digital voice in packets of 10 ms duration. It is officially described as *Coding of speech at 8 kbit/s using code-excited linear prediction speech coding* (CS-ACELP) [25]. Because of its low bandwidth requirements, G.729 is mostly used in VoIP applications where bandwidth must be conserved, such as conference calls. Standard G.729 operates at a bit rate of 8 kbit/s, but there are extensions, which provide rates of 6.4 kbit/s and 11.8 kbit/s for worse and better speech quality, respectively. However, it is a rather costly codec in terms of CPU processing time, therefore some VoIP phones and adapters can only handle one G.729 call at a time.

For audio compression was used G.729AB with 8 kbits/s bitrate and working DTX.

2.10.6 GSM-FR

GSM-Full Rate (GSM-FR) speech codec was adopted by the 3GPP for mobile telephony. The codec operates on each 20 ms frame of speech signals sampled at 8 KHz and generates compressed bit-streams with an average bit-rate of 13 kbps. The codec uses Regular Pulse Excited — Long Term Prediction — Linear Predictive Coder (RPE-LTP) technique to compress speech. The codec provides VAD (see 2.2) and comfort noise generation CNG algorithms and an inherent packet loss concealment (PLC) algorithm for handling frame erasures. The codec was primarily developed for mobile telephony over GSM networks.

2.10.7 Advanced Audio Coding

Advanced Audio Coding (AAC) is a standardized, lossy compression and encoding scheme. Designed to be the successor to the MP3 format, AAC has been standardized as part of the MPEG-2 and MPEG-4 specifications. It is widely used in YouTube, iPhone, iOS, and Android-based phones [20].

For waveform transcoding I used neroAacEnc and neroAacDec to transcode AAC8 and AAC16.

2.10.8 Speex

Speex bases its compression on CELP. It is patent-free audio compression format designed for speech. The Speex Project aims to lower the barrier of entry for voice applications by providing a free alternative to expensive proprietary speech codecs. Moreover, Speex is well-adapted to Internet applications and provides useful features that are not present in most other codecs [20].

2.10.9 MPEG-1

MPEG-1 or MPEG-2 Audio Layer III, more commonly referred to as MP3, is an audio coding format for digital audio which uses a form of lossy data compression. It is a common audio format for consumer audio streaming or storage, as well as a de facto standard of digital audio compression for the transfer and playback of music on most digital audio players. The compression works by reducing accuracy of certain parts of sound based on psychoacoustic criteria [20].

For transcoding, I used lame with 64, 32, 16 and 8 kbit/s rates.

2.10.10 Windows Media Audio

Windows Media Audio (WMA) is a compression technology developed by Microsoft. The name can be used to refer to its audio file format or its audio codecs. It is a proprietary technology that forms part of the Windows Media framework. WMA consists of four distinct codecs. The original WMA codec, known simply as WMA, was conceived as a competitor to the popular MP3 and RealAudio codecs. WMA Pro, a newer and more advanced codec, supports multichannel and high resolution audio. A lossless codec, WMA Lossless, compresses audio data without loss of audio fidelity (the regular WMA format is lossy). WMA Voice, targeted at voice content, applies compression using a range of low bit rates [20].

Chapter 3

Experimental Setup

3.1 Training Data

I used PRISM dataset [10]. PRISM contains data from all NIST SREs, beginning with the year 2005 until 2010 [1]. Also, NIST 2004, Switchboard (Phase 1 and 2 and Cellphone phase 1 and 2) and Fisher English (part 1 and 2) data are included in the dataset for training purpose. In total, PRISM dataset contains 16247 speakers and 86681 audio files.

I used 15602 audio files (1179 speakers) for UBM training, 86680 audio files (16247 speakers) for i–vector and 61206 audio files (3425 speakers) for PLDA training for the Baseline system.

3.2 Test Set and Evaluation Metric

I selected clean lapel microphone data from NIST SRE 2008 and 2010 for our experiments with noise, reverberation and codec degraded speech. I used 2002 files for PLDA/WCC training, 1088 files for enrollment/test making 2450 target trials and 592508 nontarget trials — following the division from PRISM set for making noise and reverberant conditions.

The Equal Error Rate (EER) and detection cost function (DCF) are used as a primary evaluation metric. Two DCF metrics are reported: $\text{DCF}^{\text{min}}_{\text{old}}$and $\text{DCF}^{\text{min}}_{\text{new}}$which correspond to the primary evaluation metric for the NIST speaker recognition evaluation in 2008 and 2010 respectively. The difference is that in 2010 NIST focus more on lower false alarm scenario. For more details see evaluation plans of NIST SRE [1].

3.3 Voice Activity Detection

Experiments were based on speech segments found by a voice activity detector (VAD) time alignments extracted from the clean speech and applied to the transcoded data. Therefore, all the data, clean and transcoded, contained the same sample durations. This process bypasses the problem of speech detection in transcoded data to provide an unbiased view of codec degradation on speaker recognition performance. It should be noted that some codecs change the length of the file during the encoding or decoding phase.

[1] www.itl.nist.gov/iad/mig/tests/sre/

3.4 System Description

As a speaker identification system was used system developed by **Speech@Fit** group, see Figure 3.1.

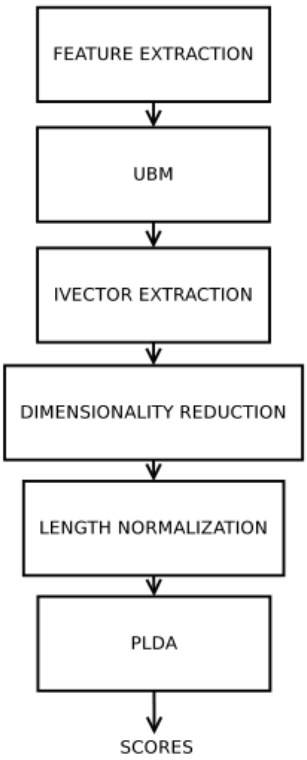


Figure 3.1: *Basic schematic description of* ***Speech@Fit*** *speaker identification system [21].*

I used 19 MFCC coefficients plus C_0 and its first and second derivatives. Bottleneck features are 80 dimensional and are trained on Fisher English Corpus [2]. The system uses 2048 Gaussian Mixture Components with diagonal covariances and i–vector with 600 dimensions. For more details see the description for the best system from [19].

[2] https://catalog.ldc.upenn.edu/LDC2004S13 and https://catalog.ldc.upenn.edu/LDC2005S13

Chapter 4

Experiments

Experiments presented in this work are based on system description in Chapter 3 and usage of techniques for improving robustness from Section 2 - PLDA and WCC respectively.

4.1 Training PLDA

I first analyzed the effect of codecs on speaker identification performance. The main subject of investigation is how the PLDA model is affected when trained with clean data, additional noisy and reverberated speech and finally additional transcoded speech.

4.1.1 Baseline

The first system can be considered as the Baseline. Here the PLDA model was trained only with clean speech data. Results are presented in Figure 4.1 in terms of Equal Error Rate (EER) when speech is degraded using each codec. For enrollment/test I used only female speakers, who usually have have lower performance, but our experiments have same trend for male speakers — our systems are gender independent. Results are presented for each of the two features described in Section 2.3.

Based on Figure 4.1, we can conclude, that several codecs result in significant EER degradation relative to clean conditions. Specifically, EER of AAC codecs is higher on both our systems. Also, *BN+MFCC* performance is much better, as I expected. Average EER on all codecs conditions is ***1.94%*** for *BN+MFCC* and ***3.94%*** for *MFCC* respectively.

4.1.2 Exposing PLDA to Noise and Reverb

I retrained the PLDA model with the clean data and additional noisy and reverberated data. Noisy data are generated from babble noise degraded speech at 8, 15, and 20 dB SNRs, I used the same PLDA training set as for codec degraded speech. The goal of this experiment is to analyze whether the PLDA model exposed to noisy and reverberated speech is more robust to codec distortions than the one trained only with clean speech. For ease of comparison, results are overlaid on the EERs from the clean PLDA model. Results are presented in Figure 4.2.

Figure 4.2 indicates that some general downward trend in EER can be observed by introducing noisy and reverberated data in the PLDA model, but also has the opposite effect on AMR codec family. We found similar feature rank ordering compared to the clean system and therefore it can be concluded that including noisy and reverberated speech in

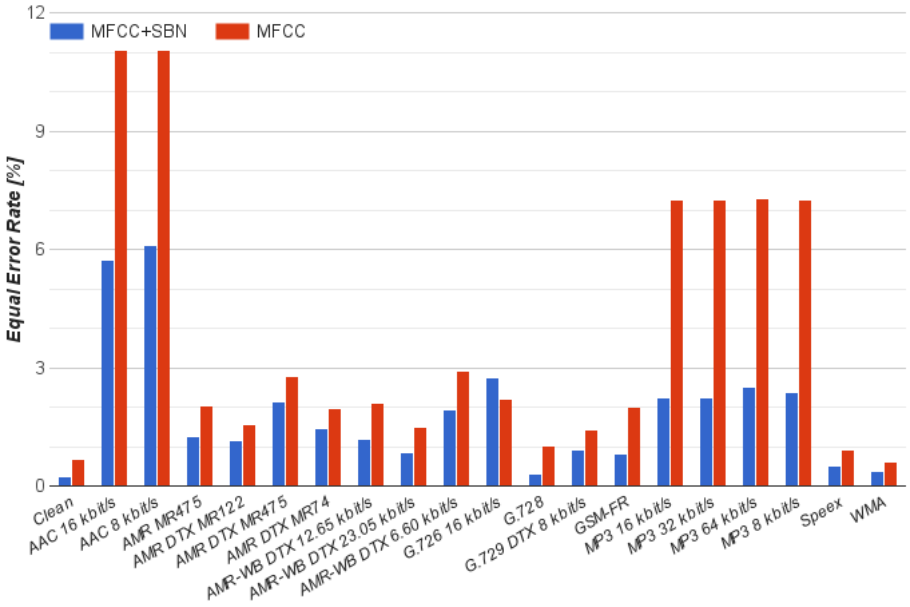


Figure 4.1: *Equal Error Rate (EER) of clean and codec-degraded evaluation data using a clean speech PLDA model and no data for WCC.*

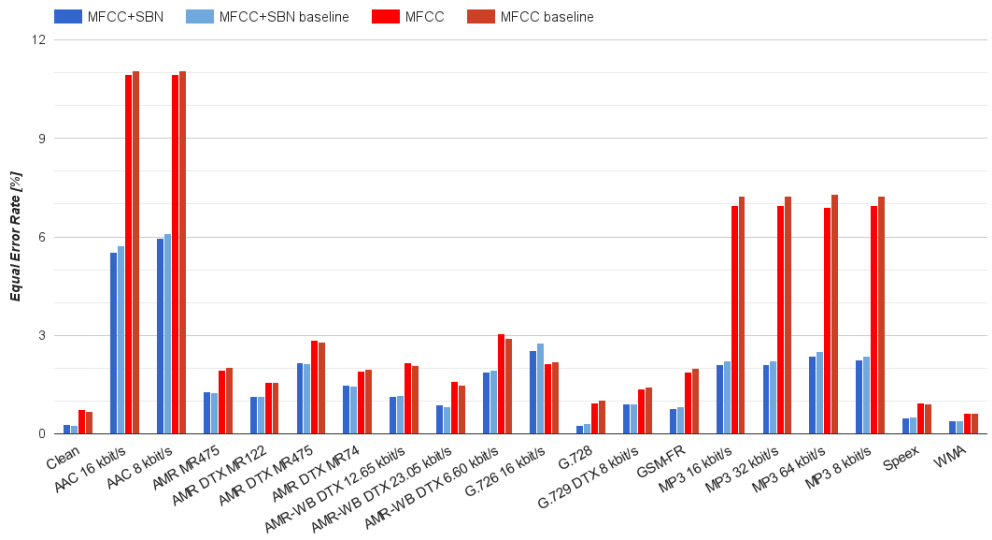


Figure 4.2: *Relative EER Improvement of clean and codec-degraded evaluation data using a PLDA model trained on clean, noisy and reverberated speech (overlaid on EER from Figure 4.1).*

the PLDA model provided some additional system robustness to codec degraded speech. Average EER on all codecs conditions is ***1.87%*** which is ***3.61%*** relative improvement comparing to Baseline for *BN+MFCC* and ***3.82%*** with relative improvement ***3.05%*** for *MFCC* respectively. This system will be denoted as ***Baseline 2*** for the rest of the paper.

4.1.3 PLDA Unseen Codec Experiments

To observe the benefit of re-training the noise and reverb PLDA model along with codec-degraded data, I included in the training set data transcoded with each codec variant except the codec used for the test. The goal is to analyze if a PLDA model exposed to multiple codec degradations other than the one used in the testing case is more robust than a PLDA training set without codec-degraded speech, for results see Figure 4.3.

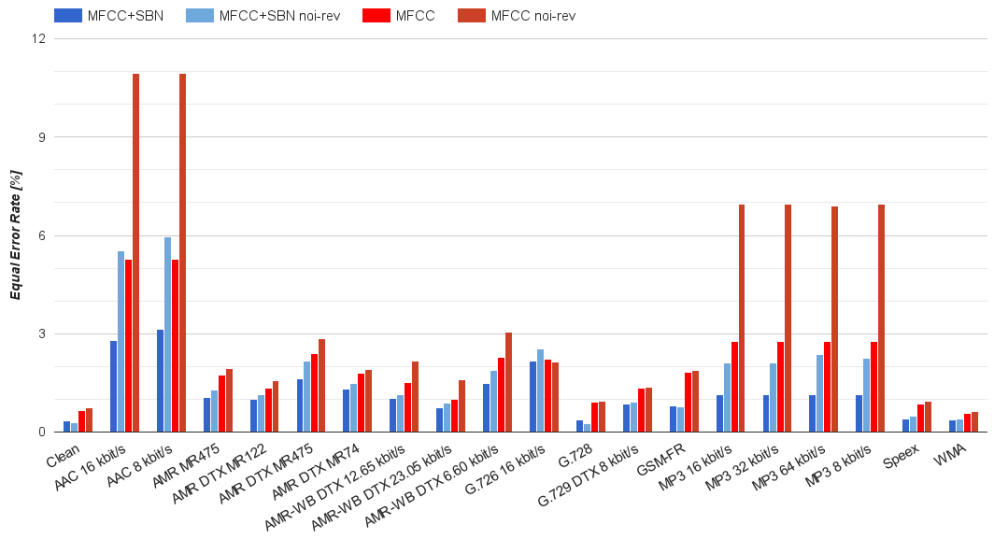


Figure 4.3: *Relative EER Improvement of clean and codec-degraded evaluation data using a PLDA models trained on clean, noisy, reverberated, and codec-degraded speech except the evaluated codec (overlaid on EER from Figure 4.2).*

As expected, from Figure 4.3 we can see significant improvement in robustness of our system, but this can be considered as optimistic case, because in many cases, we have more codec variants from one codec family. Average EER on all codecs conditions is ***1.24%*** which is ***33.69%*** relative improvement comparing to Baseline 2 for *BN+MFCC* and ***2.18%*** with relative improvement ***28.52%*** for *MFCC* respectively.

In the next experiment, I grouped the codecs by classes: AAC, AMR, AMR-WB, G.726, G.728, G.729, GSM-FR, MP3, WMA and Speex. The codec group from which the enrollment and testing came was excluded during PLDA re-training, see Figure 4.4.

According to Figure 4.4, we can clearly see improvement of robustness of our systems, PLDA model definitely provided improvement over the noise and reverberated model. Average EER on all codecs conditions is ***1.63%*** which is ***12.83%*** relative improvement comparing to Baseline 2 for *BN+MFCC* and ***3.78%*** with relative improvement ***10.47%*** for *MFCC* respectively.

4.1.4 PLDA Using All Codecs

Here, I explore the case where the PLDA model is exposed to all available codec-degraded speech along with noisy and reverberate speech. This can be considered an optimal case where the PLDA has been exposed to multiple codecs including the one used in the enrollment and test audio, see Figure 4.5.

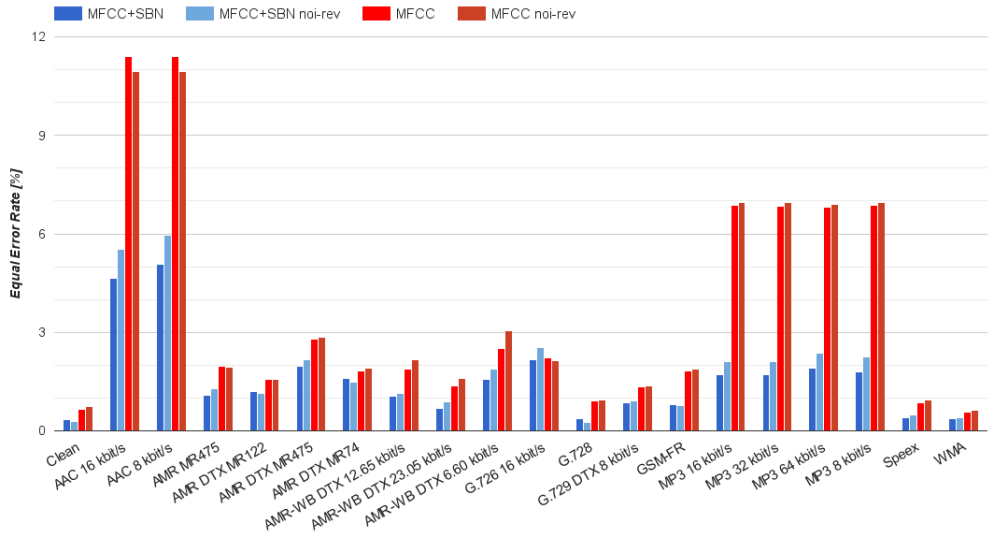


Figure 4.4: *Relative EER Improvement of clean and codec-degraded evaluation data using a PLDA models trained on clean, noisy, reverberated, and codec-degraded speech except the evaluated group (overlaid on EER from Figure 4.2).*

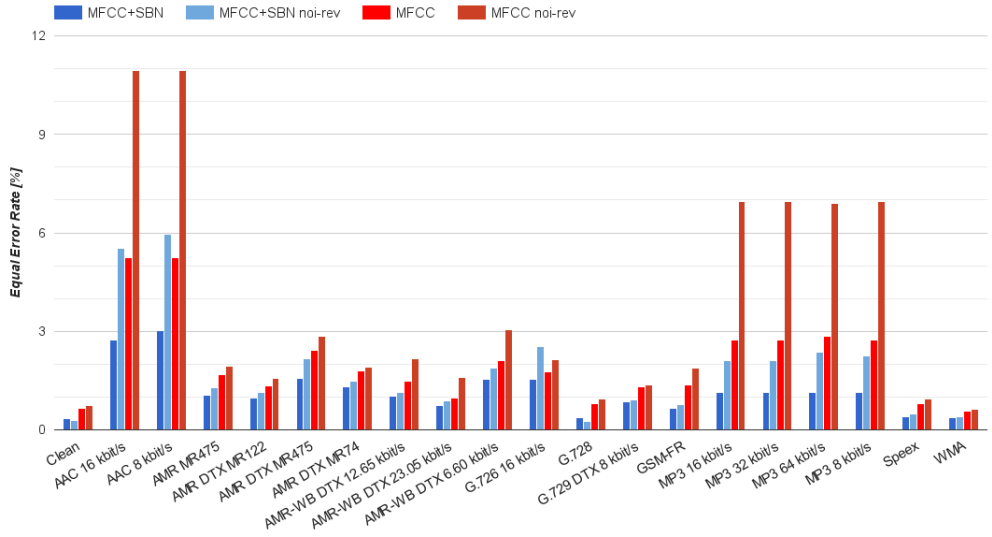


Figure 4.5: *Relative EER Improvement of clean and codec-degraded evaluation data using a PLDA models trained on clean, noisy, reverberated, and codec-degraded speech (overlaid on EER from Figure 4.2).*

Figure 4.5 indicates that including speech compressed with the same codec as used for model enrollment and testing in the PLDA training data set significantly lowered EERs. As previously observed, the PLDA model provided improved performance over MFCC in the high EER codecs: AAC and MP3. These results indicate that, unless the codec used to transcode enrollment and test data has been observed during PLDA training, the system will offer limited robustness to the degradation that transcoding imparts to the speech. Average EER on all codecs conditions is ***1.19%*** which is ***36.5%*** relative improvement

comparing to Baseline 2 for *BN+MFCC* and ***2.11%*** with relative improvement ***44.9%*** for *MFCC* respectively.

4.2 WCC and PLDA Experiments

The same experiments as I ran with PLDA and were described in Section 4.1 I ran with WCC and then with re-training PLDA model and WCC together.

4.2.1 Parameter α Estimation

Within-Class Covariance Correction (WCC), as described in 2.9 operates with constant $\boldsymbol{\alpha}$. Here, I experimentally tried to estimate it on PLDA trained on noise and reverberated data while using all codec's data for WCC. Detailed results (EER, DCF) are presented in Table 4.1.

Table 4.1: *Estimation of parameter α. Average results (EER, DCF) on all codecs conditions.*

	BN+MFCC			MFCC		
	EER	$\text{DCF}^{\text{min}}_{\text{new}}$	$\text{DCF}^{\text{min}}_{\text{old}}$	EER	$\text{DCF}^{\text{min}}_{\text{new}}$	$\text{DCF}^{\text{min}}_{\text{old}}$
0.1	0.024029	0.298206	0.106249	0.054454	0.413253	0.197328
0.2	0.024058	0.297719	0.106261	0.054506	0.413053	0.197473
0.5	0.024035	0.296683	0.106171	0.054492	0.413091	0.197489
1.0	0.024006	0.296755	0.106122	0.054537	0.412991	0.197484
2.0	0.024011	0.296683	0.105976	0.054594	0.412074	0.197500
5.0	0.023856	0.296612	0.105938	0.054668	0.412268	0.197475
10.0	0.023780	0.295791	0.106039	0.054609	0.411860	0.197650
20.0	0.024018	0.295965	0.106017	0.054566	0.411215	0.197592
50.0	0.023982	0.294670	0.105390	0.054591	0.411714	0.197744
100.0	0.024052	0.292134	0.104909	0.054597	0.411999	0.197927
1000.0	0.023920	0.292325	0.104405	0.054464	0.410308	0.197687

Based on Table 4.1, we can see, that value of constant α does not have any significant impact on our system - our system is very insensitive to this constant. The best results for both features are highlighted and will be used for the rest of this paper.

4.2.2 Unseen and All Codecs Experiments

As described before, I used both techniques — WCC and PLDA to improve our system's robustness. Table 4.2 shows the detailed results. We can see that adding different variety of transcoded speech helps also for the unseen codec degraded data. There is another gain 20% relative in average if the system is exposed in the training to the target codec data. Generally WCC technique yields slightly better results than exposing codec degraded speech data to the PLDA. There is a slight gain if we use both techniques together. Detailed results are available in Appendix B.

Table 4.2: *Average EER and DCF using previously presented techniques on all codecs conditions.*

	BN+MFCC			MFCC		
	EER [%]	DCF^{min}_{new}	DCF^{min}_{old}	EER [%]	DCF^{min}_{new}	DCF^{min}_{old}
Baseline 2	1.87	0.2496	0.0771	3.82	0.3437	0.1365
PLDA Unseen Codec	1.63	0.2434	0.0710	3.78	0.3466	0.1352
WCC Unseen Codec	1.61	0.2353	0.0695	3.73	0.3419	0.1348
WCC and PLDA Unseen Codec	1.57	0.2425	0.0704	3.75	0.3425	0.1333
PLDA Using All Codecs	1.19	0.2503	0.0833	2.11	0.2038	0.0531
WCC Using All Codecs	1.27	0.2089	0.0563	2.16	0.2627	0.0893
WCC and PLDA Using All Codecs	1.21	0.2038	0.0536	2.13	0.2459	0.0828

4.3 Robustness Focused on GSM-FR

This section describes the experiments where all training data (61206 files) are transcoded by GSM-FR codec not only 2002 files as in previous sections. By doing this we can compare the results from the system tuned for the GSM-FR codec and general system adapted to the all codecs. For this experiment I used BN+MFCC system and PLDA trained on all GSM-FR transcoded data together with noised and reverberated version as in Baseline 2.

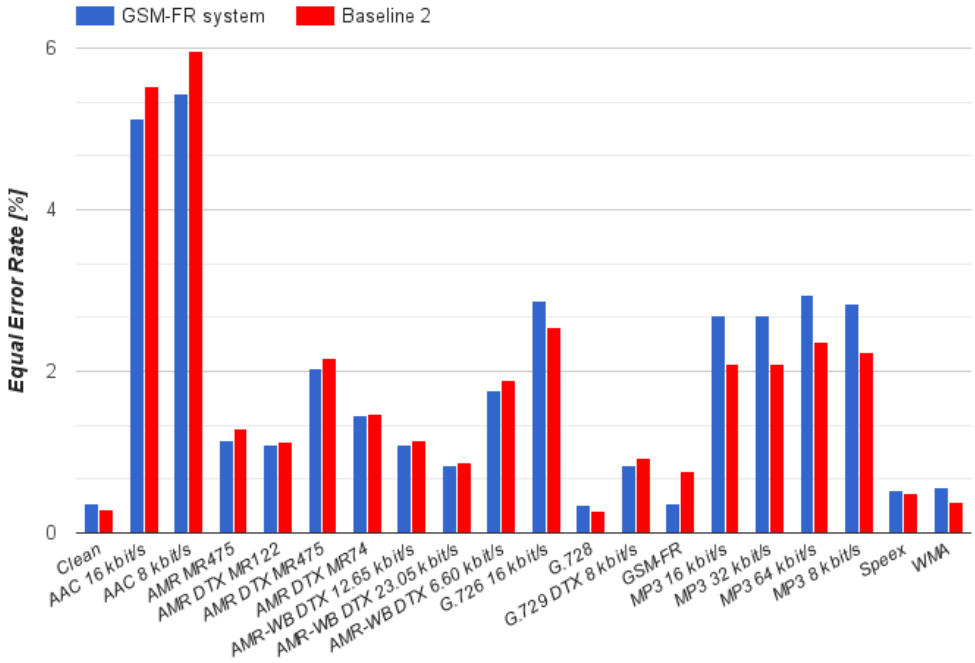


Figure 4.6: *EER of system trained on GSM-FR codec degraded data. PLDA models are trained on clean, reverberated and noised data and we are not using any data for WCC. Overlaid on EER from Figure 4.2.*

As expected, Figure 4.6 shows that there is a big gain for our GSM-FR condition, (***51%*** relative against Baseline 2). There is also very small gain for other codecs if we compare to the system which was not exposed to any, except G.726, MP3, Speex and WMA.

This scenario allow us also compare all presented techniques using this codec only. Ta-

Table 4.3: *EER and DCF of GSM-FR condition with speaker identification system using described technique for robustness improvement.*

System Description	EER [%]	DCF^{min}_{new}	DCF^{min}_{old}
Baseline	0.8278	0.1612	0.0324
Baseline 2 + Codecs in PLDA (without GSM-FR)	0.7894	0.1487	0.0360
Baseline 2 + Codecs in PLDA and WCC (without GSM-FR)	0.7802	0.1491	0.0356
Baseline 2 + GSM-FR in WCC	0.7674	0.1450	0.0317
Baseline 2	0.7557	0.1473	0.0309
Baseline 2 + Codecs in WCC (without GSM-FR)	0.7231	0.1355	0.0326
Baseline 2 + Codecs in PLDA	0.6683	0.1527	0.0352
Baseline 2 + Codecs in PLDA and WCC	0.6677	0.1525	0.0343
Baseline 2 + GSM-FR in PLDA and WCC	0.6073	0.1355	0.0326
Baseline 2 + GSM-FR in PLDA	0.6051	0.1484	0.0271
Baseline 2 + Codecs in WCC	0.6044	0.1335	0.0288
System trained with GSM-FR data - Baseline	0.4094	0.1248	0.0215
System trained with GSM-FR data - Baseline 2	0.3709	0.1199	0.0243

ble 4.3 summarizes the results. We can see that we can get up to ***27%*** relative improvement against Baseline (EER=0.83%) from system using All codecs in WCC (EER=0.60%). The specialized system for GSM-FR reaches ***55%*** relative gain against Baseline, but this system is ***34%*** relatively worse (EER=1.93%) in average on other codecs then in system which uses WCC with all codecs (EER=1.27%).

Chapter 5

Analysis of Different Lengths of Speech Utterances

Next, the influence of the lengths of speech utterances on speaker verification performance is analyzed. I used VAD to extract exact amount of speech from audio record and evaluated our system on these utterances.

5.1 Baseline System

In Table 5.1 are presented results for Baseline 2 system on *sre10c05* [1] condition (female speakers only) in terms of EER. As expected, we can see significant degradation in performance when using short utterances. Based on results, we can conclude that using longer speech utterances for test/enrollment is in most cases more sufficient, even comparing to scenario, when using same lengths for both enrollment and test.

Table 5.1: *EER [%] of sre10c05,f condition with speaker verification system using stated length of speech utterance used for enrollment/test.*

Test/Enroll	**5**	**10**	**20**	**30**	**40**	**50**	**60**	**90**	**120**	**Average**
5	19.11	14.70	11.75	10.40	10.32	9.33	9.15	8.80	8.48	11.34
10	15.45	10.91	7.71	6.78	6.61	5.65	5.31	5.13	4.65	7.58
20	13.31	9.02	5.57	4.35	4.59	3.53	3.40	3.15	2.77	5.52
30	11.97	7.83	4.61	3.58	3.76	2.81	2.59	2.31	2.06	4.61
40	12.11	8.14	5.25	4.33	4.38	3.46	3.27	3.07	2.76	5.20
50	11.00	6.80	3.78	2.91	3.16	2.04	1.98	1.82	1.55	3.89
60	10.77	6.57	3.64	2.79	3.01	1.99	1.84	1.61	1.37	3.73
90	10.24	6.16	3.39	2.51	2.68	1.80	1.75	1.43	1.19	3.46
120	9.87	5.90	3.06	2.32	2.48	1.56	1.45	1.21	1.04	3.21
Average	12.62	8.45	5.42	4.44	4.55	3.57	3.42	3.17	2.87	

[1] http://www.nist.gov/itl/

5.2 Randomly Generated Length of Speech

As previously mentioned, I extracted exact amount of speech with VAD — in these experiments, I randomly generated amount of speech in uniform distribution.

Tables 5.3 and 5.2 present results, when speaker verification system is trained on records, in which length of utterance is uniformly distributed between 20 and 160 seconds. In the first case, speech is extracted from the beginning of file (first 15 seconds are skipped) and in second case from the end of file. Based on presented results, we can see, that this approach reaches slightly better results than Baseline system from 5.1.

Table 5.2: *EER [%] of sre10c05 condition with speaker verification system using - SegmBeg15_uni20-160.*

Enroll/Test	**5**	**10**	**20**	**30**	**40**	**50**	**60**	**90**	**120**	**Average**
5	16.92	12.78	10.02	9.01	8.69	7.85	7.73	7.34	6.92	9.69
10	13.61	9.46	6.55	5.66	5.69	4.72	4.64	4.43	4.14	6.54
20	11.64	7.54	4.86	3.93	4.11	3.22	3.03	2.79	2.52	4.85
30	10.62	6.90	4.38	3.21	3.58	2.63	2.43	2.14	1.94	4.20
40	10.85	7.27	5.10	4.12	4.36	3.54	3.36	3.09	2.92	4.96
50	9.85	6.23	3.67	2.72	3.16	2.14	2.04	1.77	1.67	3.69
60	9.54	6.01	3.54	2.57	2.92	1.97	1.89	1.63	1.48	3.51
90	9.11	5.63	3.20	2.61	2.78	1.90	1.89	1.47	1.39	3.33
120	8.92	5.36	3.02	2.28	2.63	1.74	1.63	1.32	1.16	3.12
Average	11.23	7.46	4.93	4.01	4.21	3.30	3.18	2.89	2.68	

Table 5.3: *EER [%] of sre10c05,f condition with speaker verification system - SegmEnd_uni20-160.*

Enroll/Test	**5**	**10**	**20**	**30**	**40**	**50**	**60**	**90**	**120**	**Average**
5	16.62	12.60	10.02	8.91	8.79	7.94	7.67	7.40	6.95	9.65
10	13.44	9.47	6.57	5.61	5.72	4.65	4.52	4.25	3.92	6.46
20	11.57	7.58	4.95	15.57	4.11	3.06	2.98	2.66	2.45	6.10
30	10.56	6.95	4.40	3.36	3.61	2.51	2.42	2.18	1.98	4.22
40	10.80	7.41	4.99	4.18	4.34	3.40	3.29	3.02	2.83	4.92
50	9.71	6.22	3.54	2.67	3.09	2.10	1.95	1.80	1.67	3.64
60	9.54	6.12	3.44	2.65	2.97	1.97	1.89	1.63	1.48	3.52
90	8.95	5.80	3.29	2.52	2.74	1.86	1.86	1.44	1.31	3.31
120	8.70	5.54	3.03	2.33	2.61	1.70	1.60	1.28	1.11	3.10
Average	11.10	7.52	4.92	5.31	4.22	3.24	3.13	2.85	2.63	

5.3 Training on Short Utterances

To observe a benefit of adding short speech utterances into system training, I ran experiments with all lengths. Table 5.4 presents results, when speaker verification system is trained on stated length of speech utterance. Based on results, we can conclude, that in

this case, system is more accurate comparing to previous systems when using shorter (*5 – 30 seconds*) utterances. However, system trained with *5 seconds* shows exact opposite behaviour comparing to previous experiment - using more speech for enrollment/test reduces performance. This trend is not obvious when training speaker verification system is trained with longer utterances.

Table 5.4: *EER [%] of sre10c05,f condition with speaker verification system using stated length of speech segment used for enrollment — test trained on stated length.*

Train/Enroll-Test	**5**	**10**	**20**	**30**	**40**	**50**	**60**	**90**	**120**	**Average**
5	15.57	10.51	7.39	6.34	7.31	5.43	5.25	4.76	4.53	7.45
10	15.00	9.30	6.02	5.04	6.03	4.11	3.90	3.54	3.47	6.27
20	15.33	8.94	5.08	4.02	5.08	3.13	2.87	2.59	2.28	5.48
30	15.66	9.05	4.97	3.67	4.70	2.72	2.46	2.10	1.92	5.25
40	16.14	9.10	4.96	3.48	4.48	2.48	2.24	1.91	1.63	5.16
50	16.69	9.60	5.13	3.52	4.47	2.35	2.15	1.81	1.53	5.25
60	17.09	9.85	5.17	3.50	4.41	2.37	1.98	1.68	1.37	5.27
90	17.98	10.33	5.31	3.46	4.23	2.19	1.80	1.43	1.10	5.31
120	18.71	10.67	5.41	3.48	4.23	2.08	1.75	1.39	1.03	5.42
Average	16.46	9.71	5.49	4.06	4.99	2.98	2.71	2.36	2.10	

Table 5.5 shows results, when system is trained together with specified length and full length. It is important to note, that as expected, this system is more robust when using more segments of speech for test/enrollment.

Table 5.5: *EER [%] of sre10c05 condition with speaker verification system using stated length of speech segment used for enrollment — test trained on stated length together with full length.*

Train/Enroll-Test	**5**	**10**	**20**	**30**	**40**	**50**	**60**	**90**	**120**	**Average**
5	15.87	9.60	5.49	4.43	5.19	3.35	3.06	2.65	2.27	5.77
10	16.25	9.41	5.22	3.90	4.86	2.91	2.58	2.22	1.96	5.48
20	16.85	9.70	5.21	3.63	4.57	2.57	2.23	1.93	1.58	5.36
30	17.24	9.94	5.18	3.55	4.41	2.42	2.05	1.80	1.41	5.33
40	17.64	10.16	5.25	3.52	4.31	2.30	1.96	1.64	1.24	5.34
50	17.93	10.41	5.34	3.62	4.34	2.30	1.94	1.55	1.23	5.41
60	18.21	10.47	5.48	3.60	4.32	2.27	1.92	1.53	1.12	5.44
90	18.58	10.65	5.55	3.58	4.25	2.22	1.81	1.44	1.04	5.46
120	18.97	10.76	5.61	3.54	4.28	2.11	1.73	1.43	1.03	5.49
Average	17.50	10.12	5.37	3.71	4.50	2.49	2.14	1.80	1.43	

Based on previous results, we can conclude that system trained with short and short with full lengths together provides some additional robustness comparing to Baseline system and other previously analyzed scenarios. In case, when we are comparing only diagonal of our Baseline (enrollment and test lengths are same), we achieve ***5.54%*** average EER — best result in this case is ***5.16%*** from Table 5.4 when training on *40* seconds of speech.

Chapter 6

Conclusions

We analyzed the impact of codec-degraded speech on a state-of-the-art PLDA-based speaker identification system. We tested the Baseline system based on MFCC coefficient and the new state-of-the art system based on merging Bottleneck features with MFCC. We compared the systems in the matched condition (transcoded data are available during the training) and mismatched condition (data from the particular codec are not seen in training). We experimented in adding transcoded data to the PLDA and WCC and to both techniques together. We compared all techniques on one codec, GSM-FR, where we achieved relative improvement in EER 27% over the Baseline. The best optimistic results, when developing highly specialized system, we got by passing all training data through GSM-FR yielding 55% relative improvement in EER over the Baseline but in this case we see significant degradation on other conditions, which is obvious because this system is designed for GSM-FR.

We also analyzed performance on shorter utterances and compared training techniques to improve performance in this task. We retrained our system on short utterances and compared it to approach, when we used randomly generated lengths. We can conclude, that when developing system robust to length extracted from audio, training together with full and short length reaches better result comparing to other techniques used in this work.

6.1 Future Work

In future work, deeper analysis of calibration of our speaker verification framework could improve performance in a significant way. Of course, wider and more precious selection of speech codecs and VAD used on codec degraded data would make my experiments more clear and satisfying. If we used other feature extractions (LPC, PLP, PNCC or MDMC) we could try run an experiments with an i–vector fusion.

Based on experiments with different lengths of speech utterances used for evaluating our system on specific conditions and re–training whole system, in future work, we could use data from all lengths to train system and possibly use it together with full lengths.

Bibliography

[1] National institute of standards and technology. http://www.nist.gov/speech/tests/spk/index.htm.

[2] Mohamad Hasan Bahari, Mitchell McLaren, Hugo Van hamme, and David A. van Leeuwen. Speaker age estimation using i-vectors. *Engineering Applications of AI*, 34:99–108, 2014.

[3] Lukáš Burget, Oldřich Plchot, Sandro Cumani, Ondřej Glembek, Pavel Matějka, and Niko Brummer. Discriminatively trained probabilistic linear discriminant analysis for speaker verification. ICASSP, 2011.

[4] W. Davis, V. Kepuska, and H. Reddy. Voice activity detection, November 20 2003. US Patent App. 10/144,248.

[5] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. 19(4):788–798, 2011.

[6] N. Dehak, P. Kenny, et al. Front–end factor analysis for speaker verification. In *IEEE Trans. on Audio, Speech and Lang. Process.*, 2010.

[7] Robert B. Dunn Douglas A. Reynolds, Thomas F. Quatieri. *Speaker Verification Using Adapted Gaussian Mixture Models.* M.I.T. Lincoln Laboratory, Massachusetts, 2000.

[8] Anna Fedorova, Ondřej Glembek, Pavel Matějka, and Tomi Kinnunen. Exploring ANN back-ends for i-vector based speaker age estimation. In *Proc. of Interspeech*, 2015.

[9] R. Fér, P. Matějka, F. Grézl, O. Plchot, and J. Černocký. Multilingual bottleneck features for language recognition. In *Proce. of Interspeech*, pages 389–393, 2015.

[10] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot, et al. Promoting robustness for speaker modeling in the community: the prism evaluation set. *https://code.google.com/p/prism-set/*, 2012.

[11] Ondřej Glembek, Jeff Ma, Pavel Matějka, Bing Zhang, Oldřich Plchot, Lukáš Burget, and Spyros Matsoukas. *Domain Adaptation via Within-class Covariance Correction in I-vector Based Speaker Recognition Systems.* ICASSP 2014, 2014.

[12] Ondřej Glembek, Pavel Matějka, Oldřich Plchot, Ján Pešan, Lukáš Burget, and Petr Schwarz. *Migrating i-vectors Between Speaker Recognition Systems Using Regression Neural Networks.* Brno University of Technology, Speech@FIT group and Phonexia s.r.o., 2015.

[13] Martin Karafiát, Lukáš Burget, Pavel Matějka, Ondřej Glembek, and Jan Černocký. ivector-based discriminative adaptation for automatic speech recognition. In *Proc. of ASRU*, pages 152–157. IEEE Signal Processing Society, 2011.

[14] P. Kenny. Bayesian speaker verification with heavy–tailed priors. keynote presentation, Proc. of Odyssey 2010, June 2010.

[15] P. Kenny, G. Boulianne, P. Oullet, and P. Dumouchel. Joint factor analysis versus eigenchannels in speaker recognition. 15(7):2072–2084, 2007.

[16] Marcel Kockmann, Lukáš Burget, and Jan Černocký. Application of speaker- and language identification state-of-the-art techniques for emotion recognition. *Speech Communication*, 53(9):1172–1185, 2011.

[17] Lukáš Burget, IKR Slides. Gaussian distribution. `https://www.fit.vutbr.cz/study/courses/IKR/public/prednasky/02_bayesovska_teorie/bayesovska_teorie.pdf`.

[18] David González Martínez, Oldřich Plchot, Lukáš Burget, Ondřej Glembek, and Pavel Matějka. Language recognition in ivectors space. In *Proc. of Interspeech*, number 8, pages 861–864. International Speech Communication Association, 2011.

[19] Pavel Matějka, Ondřej Glembek, Ondřej Novotný, Oldřich Plchot, František Grézl, Lukáš Burget, and Jan Černocký. Analysis of dnn approaches to speaker identification. ICASSP, 2016.

[20] Mitchell McLaren, Victor Abrash, Martin Graciarena, Yun Lei, and Jan Pešan. *Improving Robustness to Compressed Speech in Speaker Recognition.* INTERSPEECH 2013, 2013.

[21] Ondřej Novotný. Adaptation of speaker recognition systems. Master's thesis, Brno University of Technology, Faculty of Information technology, 2014.

[22] S. J. D. Prince and J. H. Elder. Probabilistic linear discriminant analysis for inferences about identity. In *11th International Conference on Computer Vision*, pages 1–8, 2007.

[23] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1):19–41, 2000.

[24] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *IEEE Workshop on Automatic Speech Recognition and Understandin*, pages 55–59, Dec 2013.

[25] M. Schroeder and B. Atal. *Code-excited linear prediction (CELP): High-quality speech at very low bit rates.* Proc. IEEE ICASSP, 1985.

[26] Haizhou Li Tomi Kinnunen. *An Overview of Text-Independent Speaker Recognition: from Features to Supervectors.* University of Joensuu, 2010.

Appendices

List of Appendices

Appendix A

CD Content

- **report.pdf** - this thesis in *pdf* format
- **doc** - directory with documentation and source codes needed for compilation this thesis in LaTeX
- **codecs** - directory with files needed to transcode data for used codecs
  - **scripts** - directory with scripts sources for transcoding data
  - **lists** - directory with lists - NIST SRE 08 and NIST SRE 10 files
  - **binaries** - directory with binaries for specific codecs
- **PLDA_WCC** - directory with files needed for experiments with PLDA and WCC
  - **experiments** - directory with concrete experiments
  - **scripts** - directory with source codecs used in experiments

Appendix B

Detailed Results

B.1 WCC Experiments

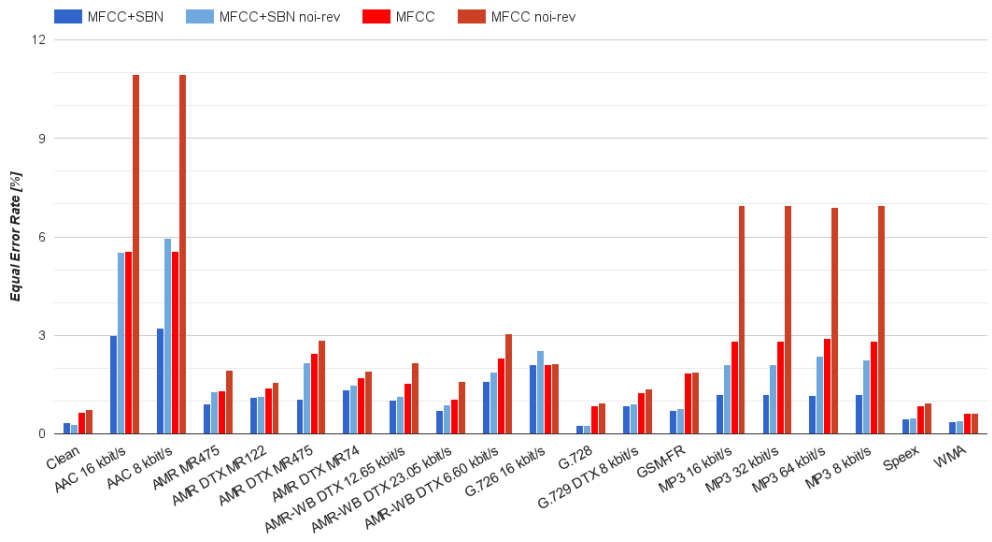


Figure B.1: *Relative EER Improvement of clean and codec-degraded evaluation data using a PLDA models trained on clean, noisy and reverberated data, while using codec-degraded speech for WCC except the evaluated codec (overlaid on EER from Figure 4.2).*

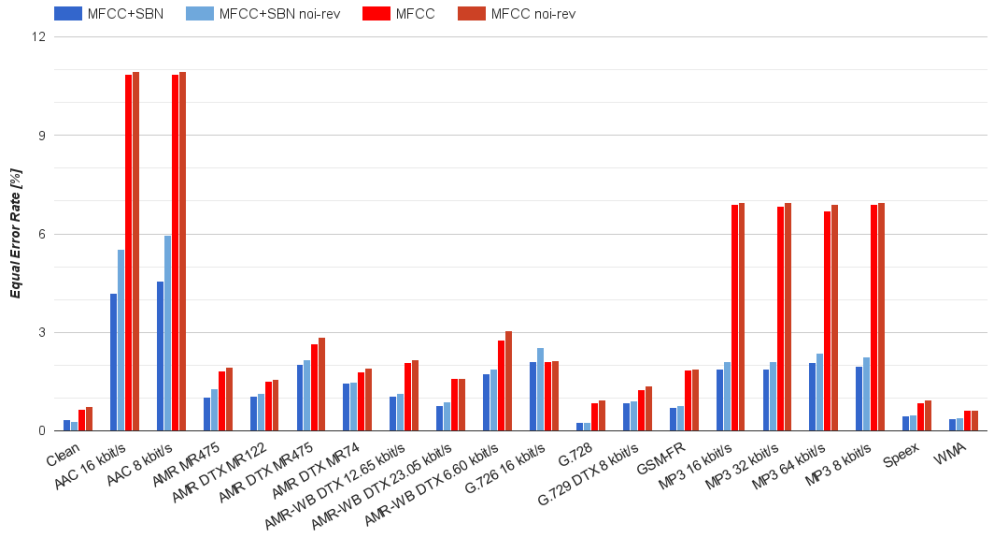


Figure B.2: *Relative EER Improvement of clean and codec-degraded evaluation data using a PLDA models trained on clean, noisy and reverberated data, while using a codec-degraded speech for WCC except the evaluated codec group (overlaid on EER from Figure 4.2).*

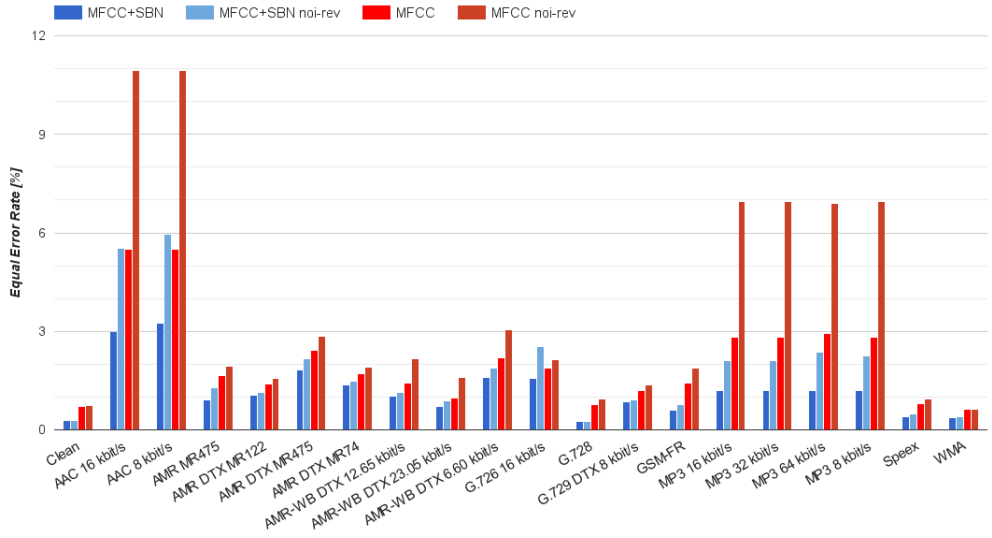


Figure B.3: *Relative EER Improvement of clean and codec-degraded evaluation data using a PLDA models trained on clean, noisy and reverberated data, while using all codec-degraded speech for WCC (overlaid on EER from Figure 4.2).*

B.2 PLDA and WCC Experiments

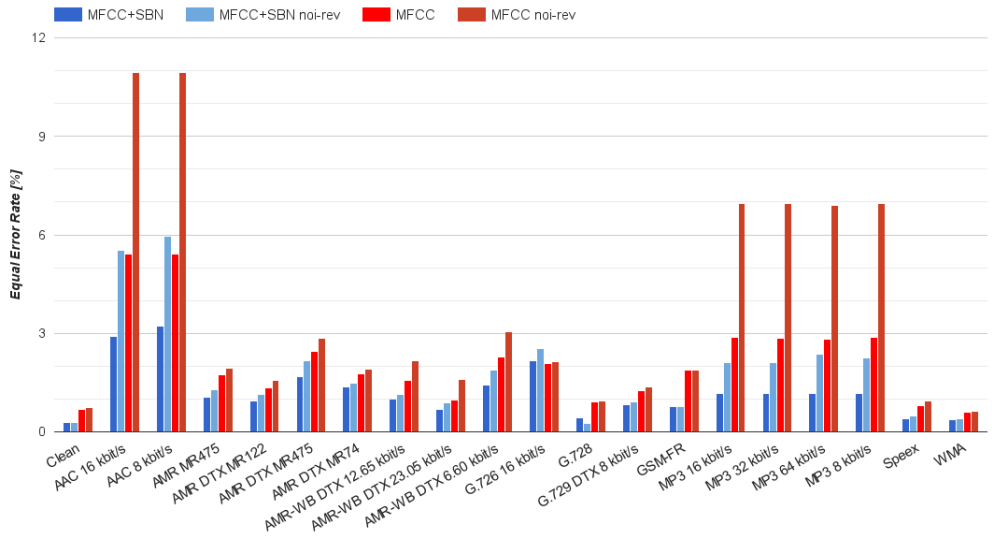


Figure B.4: *Relative EER Improvement of clean and codec-degraded evaluation data using a PLDA models trained on clean, noisy, reverberated and codec-degraded data except the evaluated codec, while using codec-degraded speech for WCC except the evaluated codec (overlaid on EER from Figure 4.2).*

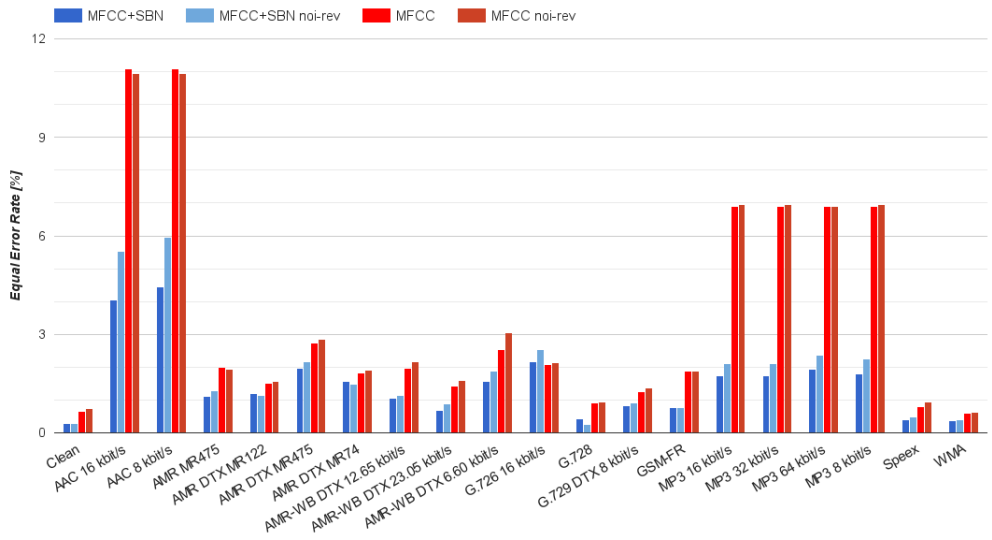


Figure B.5: *Relative EER Improvement of clean and codec-degraded evaluation data using a PLDA models trained on clean, noisy, reverberated and codec-degraded data except the evaluated codec group, while using codec-degraded speech for WCC except the evaluated codec group (overlaid on EER from Figure 4.2).*

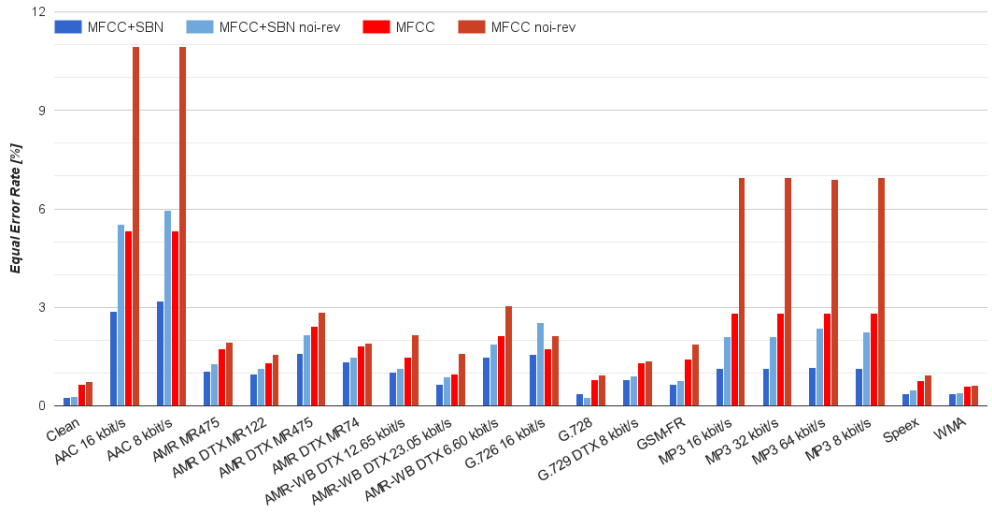


Figure B.6: *Relative EER Improvement of clean and codec-degraded evaluation data using a PLDA models trained on clean, noisy, reverberated and codec-degraded data including the evaluated codec group, while using all codec-degraded speech for WCC including the evaluated codec (overlaid on EER from Figure 4.2).*